Requested Patent: EP0898225A1

Title: APPARATUS AND METHOD FOR PROCESSING INFORMATION ;

Abstracted Patent: EP0898225 ;

Publication Date: 1999-02-24 ;

Inventor(s): MATSUURA YOKO (JP) ;

Applicant(s): SONY CORP (JP) ;

Application Number: EP19980901493 19980129 ;

Priority Number(s): WO1998JP00358 19980129; JP19970018076 19970131 ;

IPC Classification: G06F9/06 ; G06F9/445 ; G06F12/16 ;

Equivalents: WO9834169

ABSTRACT:

There are provided a non-volatile memory in which two physical sectors have been set as areas to store a basic program, an apparatus for obtaining a new basic program from the outside, and a processor for executing an application program stored in the memory in accordance with the basic program stored in the memory. The processor allows the obtained new basic program to be stored into the physical sector which is not used to write the basic program between the two physical sectors in the memory. A new basic program is stored into the other physical sector which is not the physical sector in which the basic program has been stored at the preceding time. Upon activation, the physical sector to write the basic program is switched in accordance with whether the program is normally activated or not.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 898 225 A1

(12) EUROPEAN PATENT APPLICATION

published in accordance with Art. 158(3) EPC

(43) Date of publication:
24.02.1999 Bulletin 1999/08

(21) Application number: **98901493.1**

(22) Date of filing: **29.01.1998**

(51) Int. Cl.[6]: **G06F 9/06**, G06F 9/445, G06F 12/16

(86) International application number:
PCT/JP98/00358

(87) International publication number:
WO 98/34169 (06.08.1998 Gazette 1998/31)

(84) Designated Contracting States:
DE FR GB

(30) Priority: **31.01.1997 JP 18076/97**

(71) Applicant: **SONY CORPORATION**
Tokyo 141 (JP)

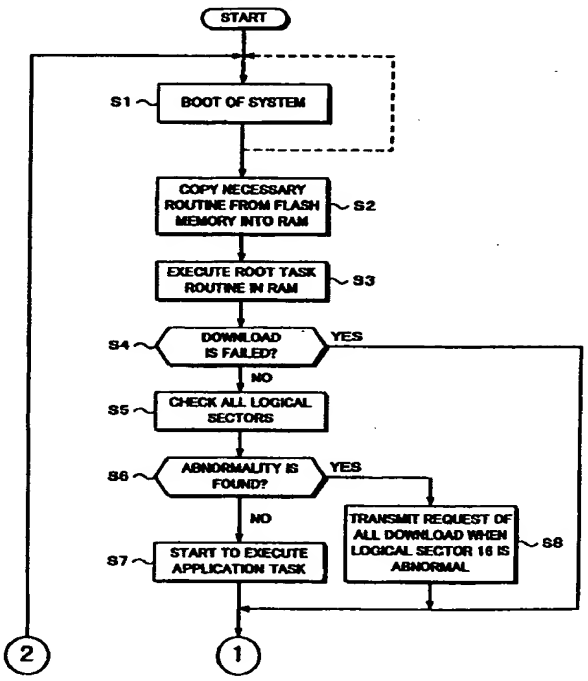
(72) Inventor: **MATSUURA, Yoko**
Shinagawa-ku, Tokyo 141 (JP)

(74) Representative:
Cotter, Ivan John et al
D. YOUNG & CO.
21 New Fetter Lane
London EC4A 1DA (GB)

(54) APPARATUS AND METHOD FOR PROCESSING INFORMATION

(57) There are provided a non-volatile memory in which two physical sectors have been set as areas to store a basic program, an apparatus for obtaining a new basic program from the outside, and a processor for executing an application program stored in the memory in accordance with the basic program stored in the memory. The processor allows the obtained new basic program to be stored into the physical sector which is not used to write the basic program between the two physical sectors in the memory. A new basic program is stored into the other physical sector which is not the physical sector in which the basic program has been stored at the preceding time. Upon activation, the physical sector to write the basic program is switched in accordance with whether the program is normally activated or not.

Fig. 4



EP 0 898 225 A1

Description

Technical Field

[0001] The invention relates to information processing apparatus and method and, more particularly, to information processing apparatus and method for rewriting a basic program which is executed upon activation and an application program which is executed in accordance with the basic program.

Background Art

[0002] In association with the progress of a recent semiconductor technique, a microcomputer is applied to various electronic equipment. For example, the microcomputer is also applied to an IRD (Integrated Receiver Decoder) which has been being widespread in recent years and serves as a receiver in a broadcasting system for broadcasting programs or the like via a satellite.

[0003] In the field which is being widespread as mentioned above, since possibility such that services to be provided will be changed in the future is strong, it is necessary to design an apparatus (for example, the IRD) so that a program of the microcomputer can be changed in correspondence to the change.

[0004] The programs of the microcomputer are classified to a basic program having routines such as a boot routine and the like for executing a basic process and an application program for actually providing services.

[0005] The applicant has already proposed such an apparatus for changing (updating) the application program in, for example, JP-A-8-195952 (1996).

[0006] In case of changing both of the basic program and the application program, when a new basic program and application program are directly written into a memory area in which original basic program and application program have been stored, a state where any basic program having a boot routine or the like is not normally stored is caused so long as an error, power failure, or the like occurs during the writing of the basic program, so that it is difficult to activate the apparatus next time. Consequently, it is considered that two storing units are provided for a rewritable non-volatile memory such as an EEPROM (Electrically Erasable and Programmable ROM) and, when the programs are updated, either one of the storing units is alternately used every updating.

[0007] In this manner, even when the error, power failure, or the like occurs during the writing of the programs, since the original basic program has been preserved, the next activation can be performed.

[0008] However, in case of providing the two storing units as mentioned above, the rewritable non-volatile memory of a capacity as much as at least that of the basic program is needed to preserve the basic program, so that there is a problem that it is difficult to reduce the costs of the apparatus.

Disclosure of Invention

[0009] The invention is made in consideration of such a situation and intends to make it possible to update programs without providing two rewritable non-volatile memories.

[0010] According to the invention of claim 1, there is provided an information processing apparatus comprising:

storing means in which at least two memory areas have been preset;
means for obtaining a new basic program from the outside; and
executing means for executing an application program stored in the storing means in accordance with a basic program stored in the storing means,
characterized in that the executing means allows a new basic program obtained to be stored into the memory area that is not used for writing the basic program in at least the two memory areas of the storing means.

[0011] According to the invention of claim 13, there is provided an information processing method characterized by comprising the steps of:

storing a basic program into one of at least two preset memory areas and executing a stored application program in accordance with the stored basic program; and
allowing a new basic program obtained from the outside to be stored into the other memory area that is not used for writing the basic program in at least the two memory areas.

[0012] According to the invention of claim 2, there is provided an information processing apparatus comprising:

storing means in which at least two memory areas to store a basic program have been preset and an application program is stored;
means for obtaining a new basic program and a new application program from the outside; and
executing means for executing the application program stored in the storing means in accordance with the basic program stored in the storing means,
characterized in that the executing means allows the obtained new basic program to be stored into the memory area that is not used for writing the basic program in at least the two memory areas of the storing means and, after that, allows the new application program to be stored into an area other than the area in which the new basic program has been stored.

[0013] According to the invention of claim 14, there is

provided an information processing method characterized by comprising the steps of:

storing a basic program into one of at least two memory areas which have been preset and executing a stored application program in accordance with the stored basic program; and
storing a new basic program obtained from the outside into the other memory area that is not used to write the basic program in at least the two memory areas of storing means and, after that, storing a new application program obtained from the outside into the area other than the area in which the new basic program has been stored.

[0014] In the information processing apparatus according to claim 1 and the information processing method according to claim 13, when the new basic program is stored into the storing means, the executing means allows the basic program to be stored into the memory area that is not used to write the basic program before. Therefore, it is possible to rewrite the basic program without any problem.

[0015] In the information processing apparatus according to claim 2 and the information processing method according to claim 14, the newly obtained basic program is stored into the memory area that is not used to write the basic program before and the newly obtained application program is written into the area other than the area in which the new basic program has been stored, so that the basic program and the application program can be stored into the storing means without any problem.

Brief Description of Drawings

[0016]

Fig. 1 is a block diagram showing an embodiment of an information processing apparatus of the invention; Fig. 2 is a block diagram showing a constructional example of a system monitoring circuit in Fig. 1; Fig. 3 is a timing chart which is used for explaining the operation of the system monitoring circuit; Fig. 4 is a flowchart for explaining the operation of the information processing apparatus of Fig. 1; Fig. 5 is a flowchart for explaining the operation of the information processing apparatus of Fig. 1; Fig. 6 is a diagram showing an example of a memory map of a flash memory in Fig. 1; Fig. 7 is a diagram showing an example of a change in a corresponding relation between physical sectors and logical sectors at the time of all downloading; and Fig. 8 is a flowchart for explaining the operation of an exceptional process of the information processing apparatus of Fig. 1.

Best Mode for Carrying Out the Invention

[0017] Fig. 1 shows a construction of an embodiment of an information processing apparatus of the invention. A CPU 1 (executing means) operates in accordance with a program (basic program or application program) stored in a flash memory 4 (storing means) or an RAM 5 (second storing means), supplies address information to each circuit via an address bus 2, and transmits or receives data corresponding to its address to/from the circuit via a data bus 3.

[0018] The flash memory 4 is a rewritable non-volatile memory, has stored a basic program such as a boot routine or the like which is executed upon activation into a physical sector (first memory area) corresponding to a logical sector 0, and has stored an application program which is executed in accordance with the basic program into a physical sector (second memory area) corresponding to logical sectors 1 to (N-1) (N indicates the number of sectors). The logical sectors are used when the CPU 1 uses them for addressing (which will be described hereinlater).

[0019] The RAM 5 temporarily stores the program or data during the process by the CPU 1.

[0020] A system monitoring circuit 7 monitors whether a download of new basic program or application program has been succeeded or not and monitors various states such as a position where the basic program has been stored in the flash memory 4 and the like (which will be explained hereinlater).

[0021] When a reset request signal is supplied from the system monitoring circuit 7, a reset generator 8 outputs a reset signal to the CPU 1.

[0022] A download program buffer 9 is constructed by, for example, a semiconductor memory such as an RAM or the like and temporarily stores a new program which is supplied by a peripheral apparatus 11.

[0023] The peripheral apparatus 11 receives a new program which is supplied via a predetermined medium or a transmission path (not shown). As an example of the peripheral apparatus, there is a receiving unit (antenna, filter, tuner, demodulator, error correcting circuit, or the like) of a digital television broadcasting such as digital satellite broadcasting, digital ground wave broadcasting, or the like. A program is a program that is necessary for the operation of the IRD.

[0024] Fig. 2 shows a constructional example of the system monitoring circuit 7. A timer signal and download information are supplied from the CPU 1 to a state monitoring circuit 21 and an address signal of (n-1) bits in an address bus signal of n bits is also supplied. A boot sector control signal B and an error signal C outputted from the state monitoring circuit 21 are supplied to an XOR (exclusive OR) circuit 22. A signal which is outputted from the XOR circuit 22 is supplied to one input of an XOR circuit 23 and is also supplied to a non-volatile RAM (NVRAM) 31 built in the state monitoring circuit 21 and the CPU 1. In the address bus (n bits)

which is supplied from the CPU 1, a predetermined address signal A of one bit is supplied to another input of the XOR circuit 23. An address signal A' which is outputted from the XOR circuit 23 is supplied to the flash memory 4 together with the address bus signal of (n-1) bits. Further, a reset request signal R outputted from the state monitoring circuit 21 is supplied to the reset generator 8. The reset signal outputted from the reset generator 8 is supplied to the CPU 1.

[0025] In the information processing apparatus in the invention, a plurality of locations are provided as a memory location where the basic program which is executed when the system is activated is stored in the flash memory 4. When all of the programs in the flash memory 4 are downloaded and rewritten, the basic program is stored at a location different from the location where the basic program used for the operation so far has been stored. With this construction, even if an error occurs during the download of the basic program and the download fails, since the basic program that can be surely normally activated remains, it is prevented that a state where the system cannot be activated due to the failure of the download occurs.

[0026] For simplicity of explanation, description will now be made hereinbelow with respect to a case where the flash memory 4 has 0 to 31 sectors and two sectors (physical sectors 0 and 16) in which the basic program can be stored are provided in the flash memory 4. The invention, however, is not limited to those numerical values.

[0027] The operation when a power source of the system monitoring circuit 7 is turned on will now be explained with reference to Fig. 2 mentioned above and a timing chart of Fig. 3.

[0028] The state monitoring circuit 21 allows the built-in NVRAM 31 to periodically store a value of the signal outputted from the XOR circuit 22, so that the state (high level or low level) of the signal outputted from the XOR circuit 22 just before the power source of the information processing apparatus such as a receiver or the like is turned off can be stored.

[0029] When the power source is turned on, the state monitoring circuit 21 reads out an output (final boot sector information) of the XOR circuit 22 which has been stored in the built-in NVRAM 31 and is generated just before the power source is turned off at the time of the preceding activation, and supplies to the XOR circuit 22. The state monitoring circuit 21 also discriminates a state of the CPU 1 from the address information of (n-1) bits and a timer signal which are supplied from the CPU 1. When it is determined that it is an erroneous state, the circuit 21 outputs a high level signal as an error signal C to the XOR Circuit 22 (Fig. 3A). Further, when the erroneous state is detected, the state monitoring circuit 21 outputs the reset request signal R to the reset request generator 8 (Fig. 3B).

[0030] When it is assumed that a low level signal has been stored in the NVRAM 31, unless the error signal C is generated, the output of the XOR circuit 22 becomes the low level signal and the low level signal is outputted to the XOR circuit 23.

[0031] In this instance, when the basic program is downloaded at the time of the preceding activation and the download fails, naturally, the CPU 1 is not normally activated and the state monitoring circuit 21 detects such a state and outputs the high level signal as an error signal C to the XOR circuit 22 as mentioned above. Consequently, the output of the XOR circuit 22 is inverted to the high level (Fig. 3D) and the high level signal is stored in the NVRAM 31 at a timing just before timing t1. The state monitoring circuit 21 outputs the error signal C and simultaneously outputs the reset request signal to the reset generator 8. The reset generator 8 generates the reset signal at timing t1 in response to the reset request signal (Fig. 3C), so that the whole system is reactivated (mesh portion in Fig. 3). When the reactivation is executed, the high level signal stored in the NVRAM 31 is supplied as a boot sector control signal B to the XOR circuit 22 (Fig. 3E) and no error signal is generated, so that the output of the XOR circuit 22 becomes the high level signal (Fig. 3D) and is supplied to the XOR circuit 23. Since the address signal A of one bit which is supplied from the CPU 1 before the error occurs has the same value as that after completion of the reactivation (Fig. 3F), the address signal A' as an output of the XOR circuit 23 before the error occurs is inverted after completion of the reactivation.

[0032] In this manner, the address which is accessed in the flash memory 4 is switched and the system is reactivated by the preserved basic program irrespective of the failure of the download.

[0033] The operation after completion of the turn-on of the power source of the information processing apparatus of Fig. 1 will now be explained with reference to flowcharts of Figs. 4 and 5. In the following explanation, although description will be made with respect to a case where the basic program has been stored in the physical sector 0, the apparatus can also similarly operate when the basic program has been stored in the physical sector 16.

[0034] In step S1, when the power source is turned on, as shown in Fig. 6, the CPU 1 refers to a reset vector stored at the head of the logical sector 0 of the flash memory 4 in which the basic program has been stored and executes a boot routine (boot() in the diagram) stored in the address shown by its value. In this instance, when an error occurs, the above-mentioned process for the reactivation is executed.

[0035] In step S2, the CPU 1 executes a copy routine (copy_prog() in the diagram), reads out a root task routine (rootTask() in the diagram) for managing the download or executing an application task, a download routine (downloader() in the diagram) for downloading a new program from the peripheral apparatus 11, and a service routine (basic ISRs in the diagram) for executing an interrupting process from the logical sector 0, and

allows them to be stored into the RAM 5 as shown in Fig. 6.

[0036] In step S3, the CPU 1 starts the execution of the root task routine in the RAM 5. The processes in step S3 and subsequent steps are executed in accordance with the root task routine.

[0037] As mentioned above, the CPU 1 allows the routines such as a root task routine and the like stored in the logical sector 0 to be stored into the RAM 5 and, after that, executes the routines stored in the RAM 5, so that the logical sector 0 in the flash memory 4 can be rewritten. Even when a bus error, an address error, an illegal command, or the like occurs during the execution of the program, the exceptional process can be executed by storing the service routine to execute the interrupting process into the RAM 5, so that a download of a proper program can be executed after completion of the exceptional process.

[0038] In step S4, the CPU 1 inquires of the peripheral apparatus 11 (or system monitoring circuit 7) about whether the preceding download of the program from the peripheral apparatus 11 has failed or not. When it is determined that the preceding download is successful, the processing routine advances to step S5 and all of the logical sectors are checked. In case of checking the logical sectors, it is desired to sequentially check the logical sectors from the logical sector 16 serving as a rewriting start sector of the downloaded program to the logical sector 32 and, after that, sequentially check the logical sectors from the logical sector 0 to the logical sector 15. Even when the download is never executed until this time point after the power-on, the processing routine advances to step S5.

[0039] In step S6, the CPU 1 discriminates whether an abnormality has been found in all of the logical sectors or not. When it is decided that no abnormality is not found, the processing routine advances to step S7, the application tasks (application program) (applicationTask1(), applicationTask2(), ... in the diagram) are sequentially started as a multi-task, after that, the processing routine advances to step S9 (Fig. 5), and the CPU 1 waits until the download is executed.

[0040] For example, a program version register or a register for check sum is assured in a predetermined memory area in the flash memory 4, a version of the program and a value of the check sum are previously stored upon writing of the program and those values are referred in step S6, so that the CPU 1 can discriminate whether each logical sector is abnormal or not. Even when information of the result (success or failure) of the download is not stored, the peripheral apparatus 11 can determine whether the preceding download has been successful or not by checking the predetermined logical sector as mentioned above.

[0041] On the other hand, when the CPU 1 decides that the abnormality was found in step S6, the processing routine advances to step S8. In step S8, the CPU 1 discriminates whether the logical sector 16 is abnormal or not. When the logical sector 16 is abnormal, the CPU 1 determines that the writing of the boot sector failed in the preceding all download, outputs a request to download (all download) a new program corresponding to all of the logical sectors to the peripheral apparatus 11. After that, the processing routine advances to step S9 (Fig. 5) and the CPU 1 waits until the download is executed. When the logical sector 16 is normal, the processing routine advances to step S9 without executing any process.

[0042] In step S4, when it is decided that the preceding download failed, (even if the application task is executed, since a possibility such that the bus error, address error, illegal command, or the like occurs is strong,) the processing routine advances to step S9 (Fig. 5) and the CPU 1 waits until the next download is executed.

[0043] The CPU 1 waits until the next download is executed in step S9. When the download is started by the peripheral apparatus 11, first in step S10, the CPU 1 specifies the sector (boot sector) in which the boot routine has been arranged, namely, the number of the physical sector corresponding to the logical sector 0 (in this case, 0 or 16) from, for example, final boot sector information stored in the non-volatile memory 31.

[0044] When the application program is operated in a multi-task environment, the root task (having a rewriting program) is remained at this time point and the execution of all of the other tasks is stopped.

[0045] In step S11, the CPU 1 discriminates whether the kind of download to be executed is the all download or a partial download such that the programs (namely, the application program) of the sectors except for the boot sector are downloaded by a predetermined signal from the peripheral apparatus 11. When this download is the partial download, the processing routine advances to step S12 and the CPU 1 allows the program supplied from the peripheral apparatus 11 to be stored into the area in a range from the logical sector 1 to the final logical sector.

[0046] As shown in Fig. 6, the partial download can be executed by storing the application tasks and data corresponding to them into the sectors different from that of the basic program such as a boot routine and the like. In the partial download, since the basic program is not rewritten, data is protected. Therefore, even if an accident occurs during the download, the partial download can be executed again by using such a construction.

[0047] In this instance, when the boot sector is arranged in only the physical sector 0 and the activation is executed by the basic program stored in it, the program is overwritten onto the area from the physical sector 1 to the final physical sector (physical sector N-1).

[0048] On the other hand, there is considered a case where although the system is activated by the basic program stored in the physical sector 0 at present, the other boot sector is arranged in the physical sector 16. Although the all download was executed at the preced-

ing activation and the download has been normally executed up to the boot sector, if the download of the application program fails, such a state is caused. As a countermeasure in this case, there are considered a case where the boot sector of the physical sector 0 is remained and the partial download is performed to the other portions and a case where the boot sector of the physical sector 16 is remained and the partial download is performed to the other portions. As for the selection of either one of the two cases, for example, it is also possible to compare the version information of the boot sectors stored in the physical sectors 0 and 16 and to leave the new one. It is also possible to leave the boot sector having a high affinity for the program to be partially downloaded from now on and to download the other portions.

[0049] In step S13, the CPU 1 discriminates whether this download failed or not. When the download is successful, the processing routine is returned to step S1 and the system is booted again (reactivation is executed).

[0050] On the other hand, when it is decided in step S13 that the download failed, the CPU 1 allows information of the failure of the download to be stored into the built-in memory. Further in step S14, the CPU 1 reports the failure of the download to the peripheral apparatus 11 by outputting a predetermined signal. After that, the processing routine is returned to step S9 and the CPU 1 waits until the download is executed again.

[0051] When it is determined in step S11 that the present download is the all download, the processing routine advances to step S15 and the CPU 1 allows the program supplied from the peripheral apparatus 11 to be stored. That is, the boot sector is written into the logical sector 16 and data of the logical sector 1 is written into the logical sector 17. In this manner, data is sequentially written. The sector is returned to the logical sector 0. Data is written into the areas up to the logical sector 15.

[0052] As shown in Fig. 7, the addressing of the flash memory 4 is performed by, for example, 32 bits. In case of using the value of the 21th bit of the address information for the switching control of the boot sector, when the CPU 1 addresses the head (0 x 200000 ("0 x" indicates the hexadecimal number)) of the logical sector 0, the head (0 x 200000) of the physical sector 0 is addressed so long as the value of the address signal is equal to 0. On the other hand, when the value of the address signal is equal to 1, the head (0 x 300000) of the physical sector 16 is addressed.

[0053] Therefore, when the boot sector is arranged in the physical sector 0, as shown in Fig. 7, the program is sequentially written into the physical sectors from the sector 16 to the end sector (physical sector N-1) and the physical sectors 0 to 15 in accordance with the order of the logical sector numbers. Therefore, the physical sector number corresponding to a predetermined logical sector number at this time is shown by the remainder ((logical sector No. + 16)(mod N)) of the sector number N for the number obtained by adding 16 to the logical sector number.

[0054] When the boot sector is arranged in the physical sector 16, the program is sequentially written into the physical sectors from the sector 0 to the end sector (physical sector N-1) in accordance with the order of the logical sector numbers. Therefore, the number of each physical sector after completion of the writing is the same number as that of the logical sector corresponding to the physical sector.

[0055] That is, when the original boot sector is arranged to the physical sector 0, the boot sector is arranged to the physical sector 16 by the present download. When the original boot sector is arranged to the physical sector 16, the present boot sector is arranged to the physical sector 0. As mentioned above, by preventing that the program of the present boot sector is written before a new program is overwritten into the original boot sector, even when an error occurs during the download of the program, at least one normal boot sector is preserved.

[0056] In step S16, the CPU 1 discriminates whether the download has failed or not. When the download is successful, the processing routine is returned to step S1 (Fig. 4) and the system is booted again (reactivation is executed).

[0057] When it is decided in step S16 that the download failed, the CPU 1 allows the information of the failure of the download to be stored into the built-in memory. Further in step S17, after the failure was reported to the system monitoring circuit 7 and peripheral apparatus 11, the processing routine is returned to step S9 and the CPU 1 waits until the download is executed again.

[0058] In this manner, when the all download or partial download is executed as necessary and the download does fail, a predetermined application task is executed after completion of the boot.

[0059] An exceptional process when a bus error, an address error, an illegal command, or the like occurs will now be explained by referring to a flowchart of Fig. 8. In above-mentioned step S5, when an error occurs due to a check omission in spite of a fact that all of the logical sectors were checked, the following exceptional process is executed.

[0060] First in step S21, the CPU 1 reads out the address in which the error occurred and, in step S22, specifies the logical sector in which the error occurred from its address value.

[0061] In step S23, the CPU 1 discriminates whether the logical sector is the boot sector (namely, logical sector 0) or not. When the logical sector is the boot sector, in step S24, the CPU 1 transmits a request for the all download to the peripheral apparatus 11. When the sector is a sector except for the boot sector, the CPU 1 transmits a request for the partial download.

[0062] By executing the exceptional process as men-

tioned above, for example, even when the power failure occurs during the download and the failure of the download is not reported to the peripheral apparatus 11 and, further, a check omission occurs at the time of checking of the logical sectors, the download can be executed again.

[0063] In the embodiment, although the number of the physical sector to which the boot sector is arranged is set to 0 or 16, the boot sector can be also arranged to the other physical sector.

[0064] In the embodiment, although the basic program has been stored in the RAM 5, it is also possible to construct in a manner such that the logical sectors 16 to (N-1) are rewritten by a program rewriting module (downloader() in Fig. 6) stored in the boot sector (logical sector 0) in the flash memory 4 and, as for the rewriting of the logical sectors 0 to 15, the execution is shifted to a program rewriting module which is newly supplied and all of the programs are rewritten. Therefore, it is unnecessary to store the basic program into the RAM 5.

[0065] In this case, an address into which the program rewriting module (downloader()) which is newly supplied is stored is previously supplied as data to the original program rewriting module. After a new basic program was written into the logical sector 0 in accordance with the original program rewriting module, the CPU 1 shifts the execution of the program to the new program rewriting module in the new logical sector 0 and writes the remaining program subsequent to the logical sector 0 into the flash memory 4 in accordance with the new program rewriting module.

[0066] As mentioned above, according to the invention, the new basic program is stored into a predetermined area in the second memory area in which the original application program has been stored and, after that, the new application program is stored into the area in a range from the end of the area in which the new basic program has been stored to the end of the second memory area, the first memory area in which the original basic program has been stored, and the area in a range from the head of the second memory area to the head of the area in which the new basic program has been stored. Consequently, a capacity of the rewritable non-volatile memory can be reduced.

1:	CPU
2:	ADDRESS BUS
3:	DATA BUS
4:	FLASH MEMORY
5:	RAM
7:	SYSTEM MONITORING CIRCUIT
8:	RESET GENERATOR
9:	DOWNLOAD PROGRAM BUFFER
11:	PERIPHERAL APPARATUS
21:	STATE MONITORING CIRCUIT
22,23:	XOR CIRCUIT
31:	NON-VOLATILE RAM

Claims

1. An information processing apparatus comprising:

   storage means in which at least two memory areas have been preset;
   means for obtaining a new basic program from the outside; and
   executing means for executing an application program stored in said storage means in accordance with said basic program stored in said storage means,
   characterized in that said executing means allows said obtained new basic program to be stored into the memory area which is not used to write the basic program between said at least two memory areas in said storage means.

2. An information processing apparatus comprising:

   storage means in which at least two memory areas to store a basic program have been preset and an application program is stored;
   means for obtaining a new basic program and a new application program from an outside; and
   executing means for executing the application program stored in said storage means in accordance with the basic program stored in said storage means,
   characterized in that said executing means allows said new basic program obtained to be stored into the memory area which is not used to write the basic program between said at least two memory areas in said storage means and, after that, allows said new application program to be stored into the area other than the area in which said new basic program has been stored.

3. An information processing apparatus according to claim 1 or 2, characterized in that

   said storage means is a non-volatile memory.

4. An information processing apparatus according to claim 1 or 2, characterized in that

   said means for obtaining the basic program from the outside is receiving means of a digital television signal.

5. An information processing apparatus according to claim 1 or 2, characterized in that

   said apparatus further comprises second storage means for temporarily storing said basic program, and
   said executing means allows said new basic

... program and said new application program to be stored in accordance with said basic program stored in said second storage means.

6. An information processing apparatus according to claim 2, characterized in that

said executing means allows said new basic program to be stored in accordance with said basic program stored in said second storage means and, after that, allows said new application program to be stored in accordance with said new basic program.

7. An information processing apparatus according to claim 2, characterized in that

said application program has routines corresponding to a predetermined number of functions and
data which is used by said routine is stored into a sector in which said routine is stored.

8. An information processing apparatus according to claim 2, characterized in that

said executing means allows said new basic program and said new application program to be stored or allows only said new program to be stored into said storage means in accordance with a predetermined signal.

9. An information processing apparatus according to claim 1 or 2, characterized in that

each of said at least two memory areas in said storage means is one predetermined sector among a predetermined number of sectors.

10. An information processing apparatus according to claim 1 or 2, characterized in that

said basic program has a boot routine.

11. An information processing apparatus according to claim 1 or 2, characterized in that

said basic program has a routine for an interrupting process.

12. An information processing apparatus according to claim 1 or 2, characterized in that

said apparatus further has monitoring means for monitoring whether said executing means is normally activated or not, and
when said executing means is not normally activated, a write address is switched from one of said at least two memory areas in said storage means to the other and said executing means is again activated.

13. An information processing method characterized by comprising the steps of:

storing a basic program into one of at least two preset memory areas and executing a stored application program in accordance with said stored basic program; and
storing a new basic program obtained from an outside into the other memory area which is not used to write the basic program between said at least two memory areas.

14. An information processing method characterized by comprising the steps of:

storing a basic program into one of at least two preset memory areas and executing a stored application program in accordance with said stored basic program; and
storing a new basic program obtained from an outside into the other memory area which is not used to write the basic program between said at least two memory areas in said storage means and, after that, storing a new application program obtained from the outside into the area other than the area in which said new basic program has been stored.

Fig. 1

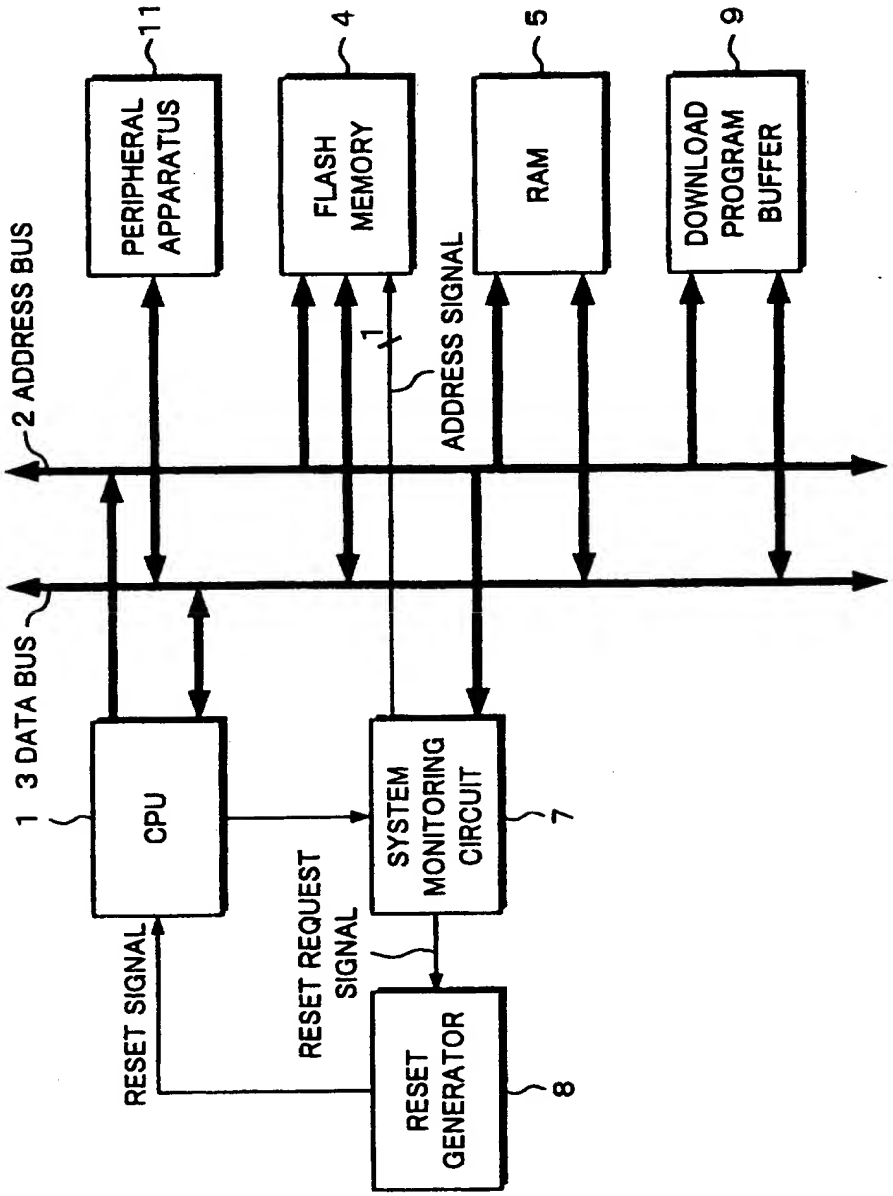
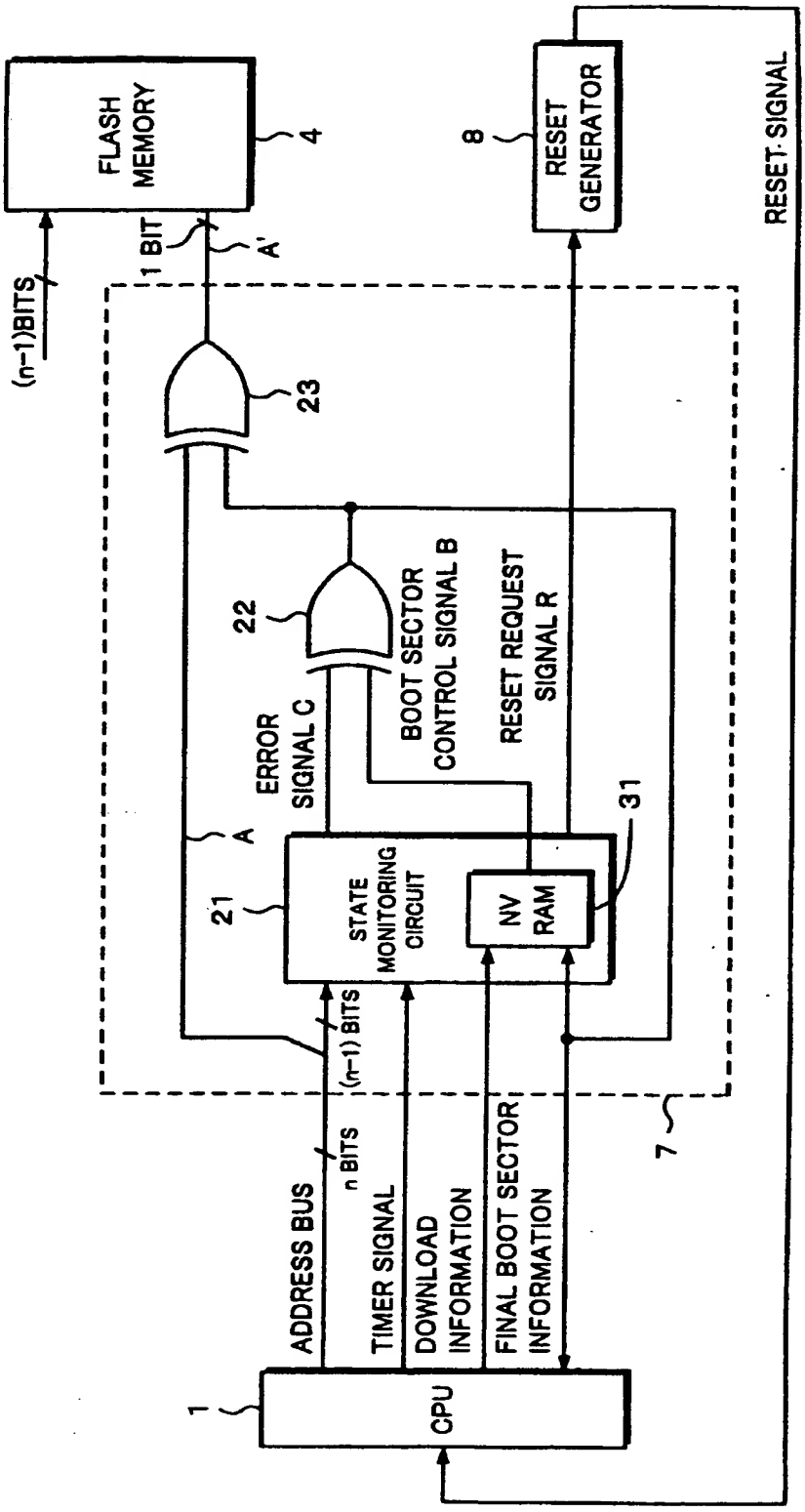

PERIPHERAL APPARATUS
11
FLASH MEMORY
4
RAM
5
DOWNLOAD PROGRAM BUFFER
9
2 ADDRESS BUS
1
ADDRESS SIGNAL
1 3 DATA BUS
CPU
SYSTEM MONITORING CIRCUIT
7
RESET REQUEST SIGNAL
RESET SIGNAL
RESET GENERATOR
8

Fig. 2


FLASH MEMORY
4
8
RESET GENERATOR
RESET SIGNAL
(n-1)BITS
1 BIT
A'
23
22
BOOT SECTOR CONTROL SIGNAL B
RESET REQUEST SIGNAL R
ERROR SIGNAL C
A
21
STATE MONITORING CIRCUIT
NV RAM
31
(n-1) BITS
n BITS
ADDRESS BUS
TIMER SIGNAL
DOWNLOAD INFORMATION
FINAL BOOT SECTOR INFORMATION
7
1
CPU

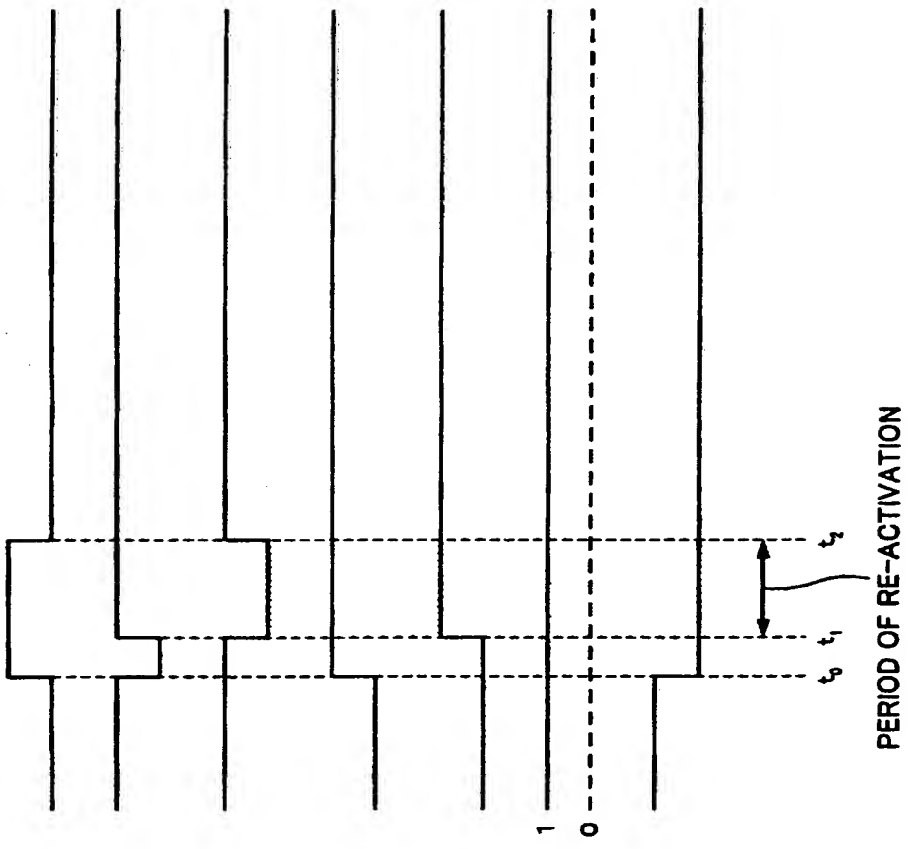


Fig. 3A

Fig. 3B

Fig. 3C

Fig. 3D

Fig. 3E

Fig. 3F

Fig. 3G

Fig. 4

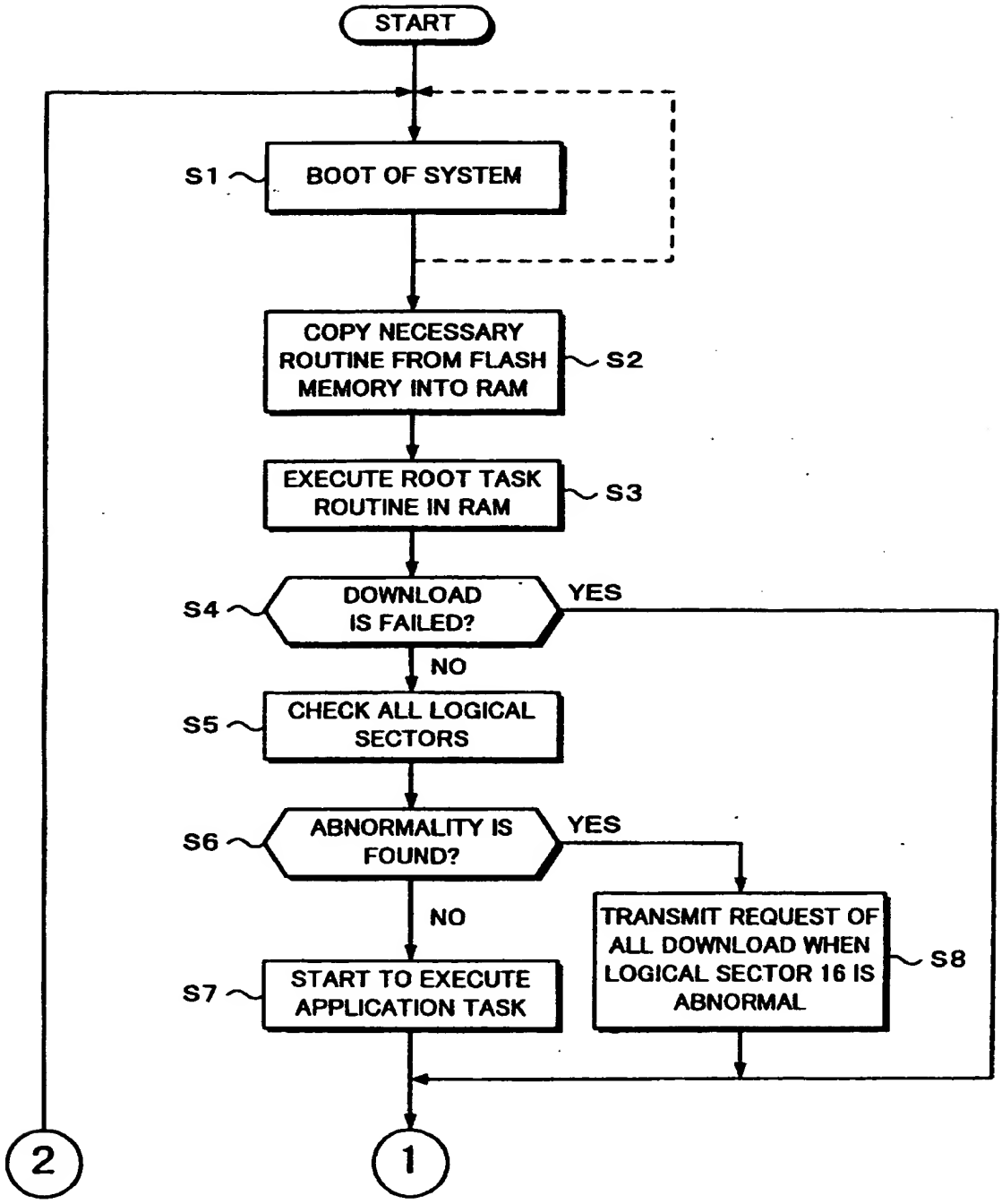

START
S1
BOOT OF SYSTEM
COPY NECESSARY ROUTINE FROM FLASH MEMORY INTO RAM
S2
EXECUTE ROOT TASK ROUTINE IN RAM
S3
S4
DOWNLOAD IS FAILED?
YES
NO
S5
CHECK ALL LOGICAL SECTORS
S6
ABNORMALITY IS FOUND?
YES
NO
S7
START TO EXECUTE APPLICATION TASK
TRANSMIT REQUEST OF ALL DOWNLOAD WHEN LOGICAL SECTOR 16 IS ABNORMAL
S8
2
1

Fig. 5

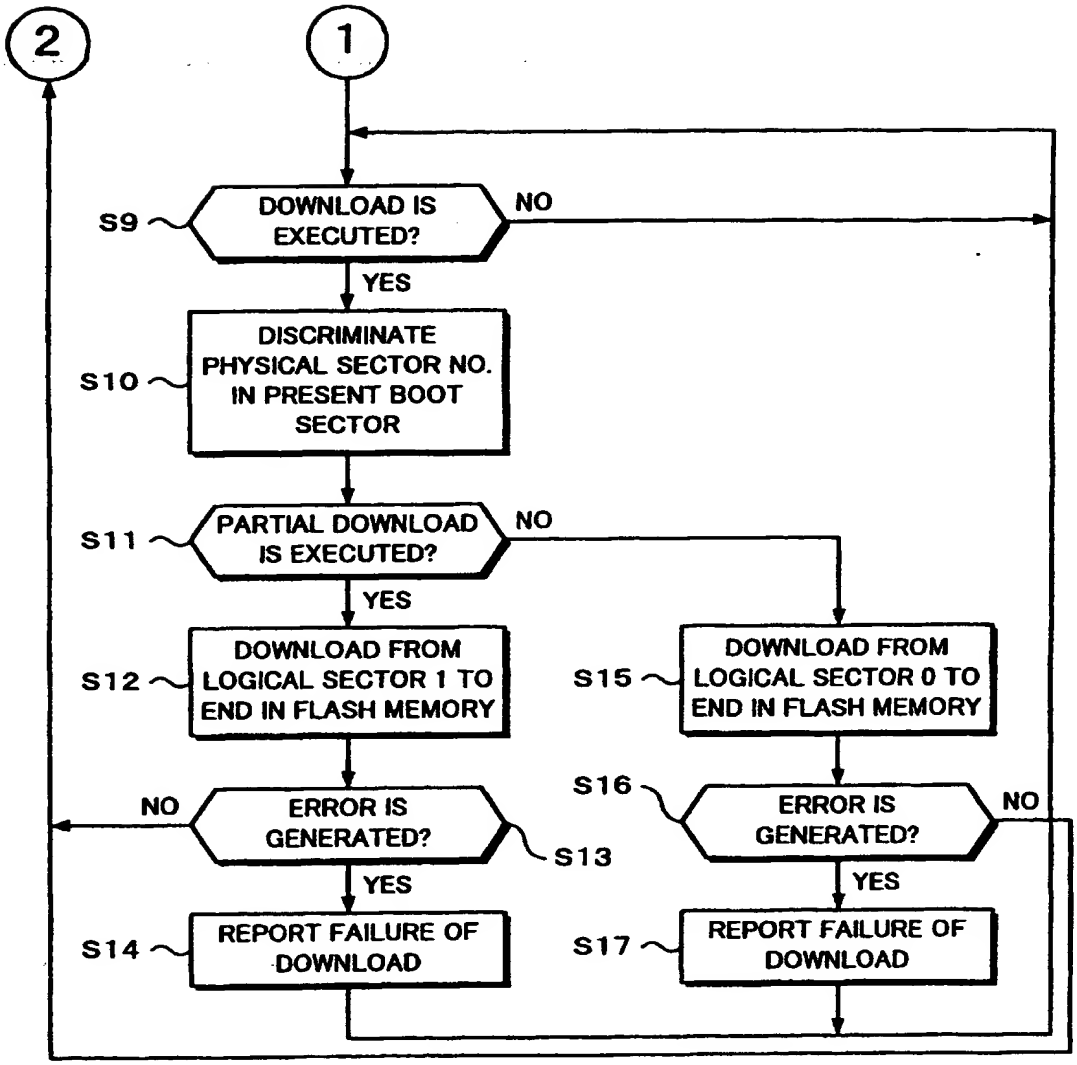

2
1
S9
DOWNLOAD IS EXECUTED?
NO
YES
S10
DISCRIMINATE PHYSICAL SECTOR NO. IN PRESENT BOOT SECTOR
S11
PARTIAL DOWNLOAD IS EXECUTED?
NO
YES
S12
DOWNLOAD FROM LOGICAL SECTOR 1 TO END IN FLASH MEMORY
S15
DOWNLOAD FROM LOGICAL SECTOR 0 TO END IN FLASH MEMORY
S16
NO
ERROR IS GENERATED?
S13
NO
ERROR IS GENERATED?
YES
YES
S14
REPORT FAILURE OF DOWNLOAD
S17
REPORT FAILURE OF DOWNLOAD

Fig. 6

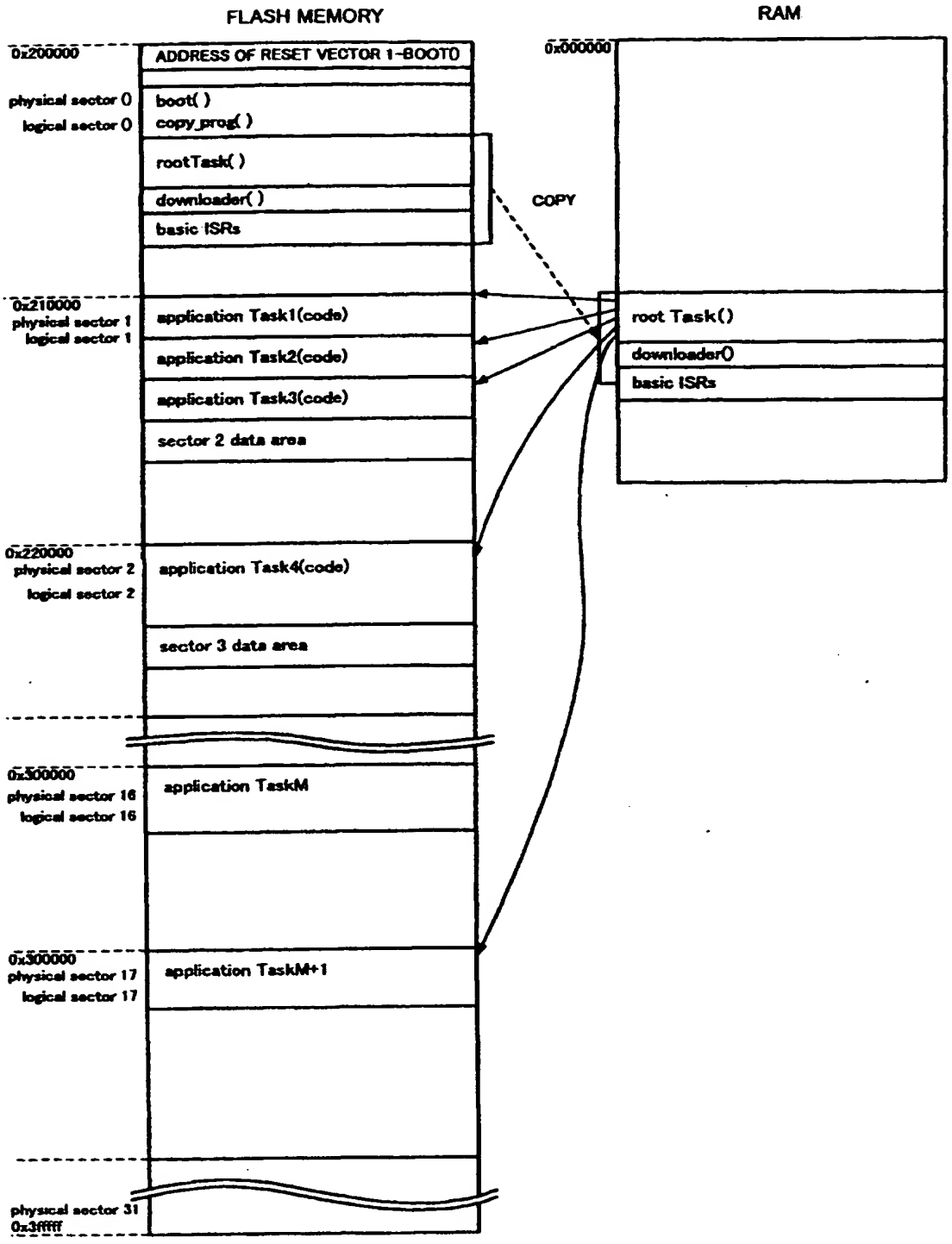

FLASH MEMORY
RAM
0x200000
ADDRESS OF RESET VECTOR 1-BOOT0
physical sector 0
logical sector 0
boot( )
copy_prog( )
rootTask( )
downloader( )
basic ISRs
COPY
0x000000
0x210000
physical sector 1
logical sector 1
application Task1(code)
application Task2(code)
application Task3(code)
sector 2 data area
root Task()
downloader()
basic ISRs
0x220000
physical sector 2
logical sector 2
application Task4(code)
sector 3 data area
0x300000
physical sector 16
logical sector 16
application TaskM
0x310000
physical sector 17
logical sector 17
application TaskM+1
physical sector 31
0x3fffff

Fig. 7

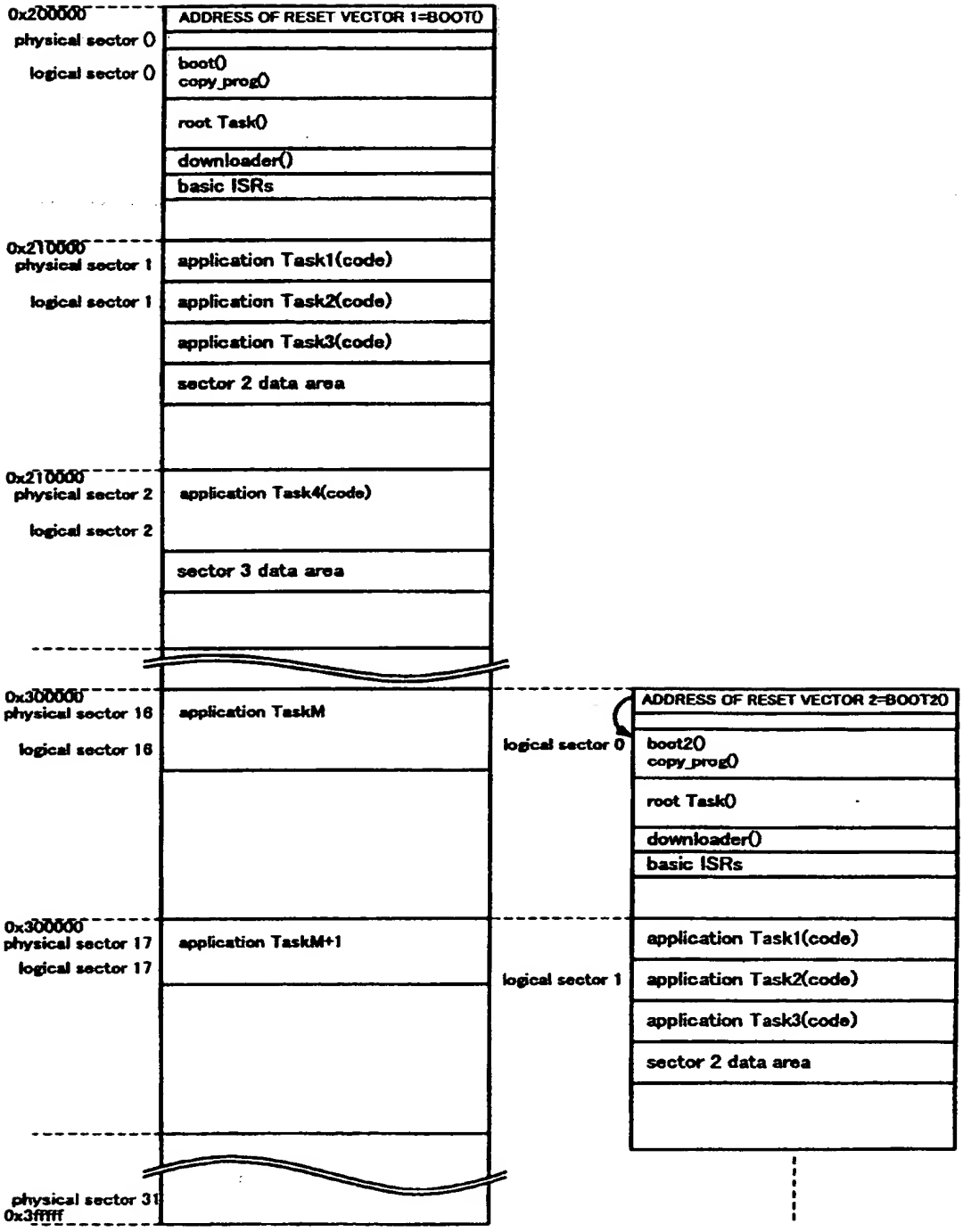
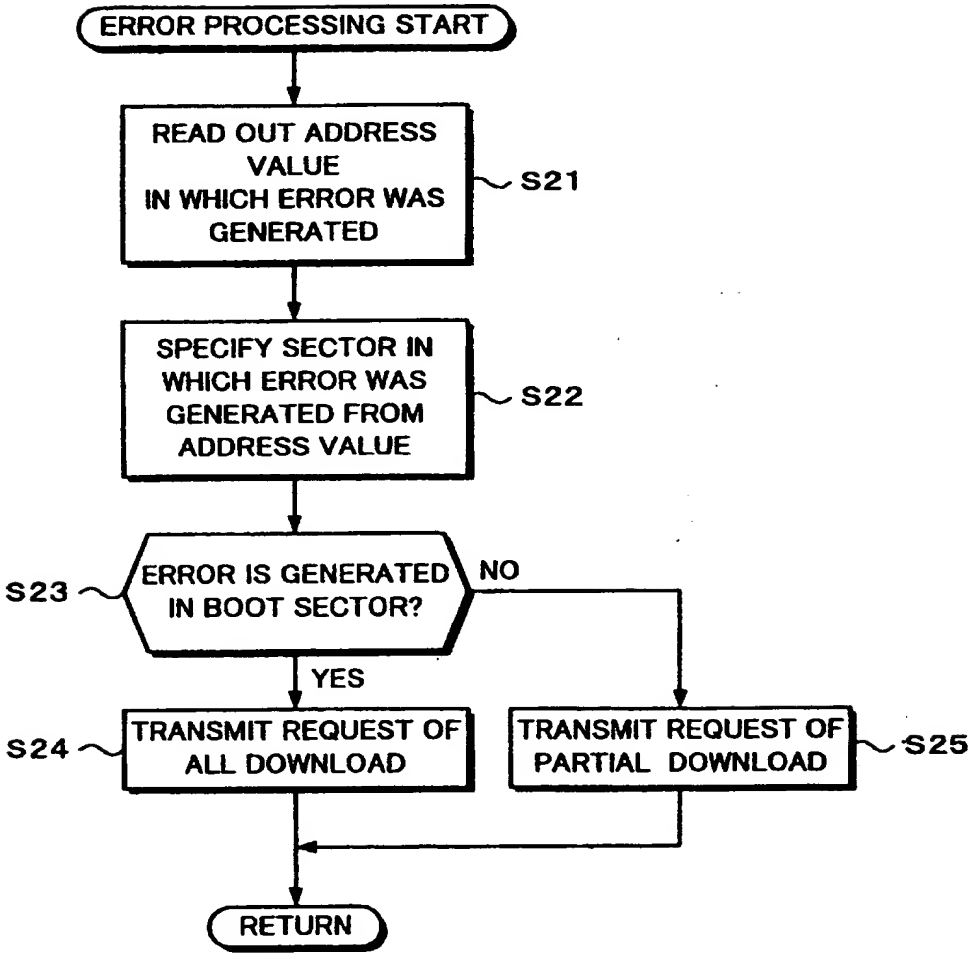

0x200000
physical sector 0
logical sector 0
ADDRESS OF RESET VECTOR 1=BOOT0
boot()
copy_prog()
root Task()
downloader()
basic ISRs
0x210000
physical sector 1
logical sector 1
application Task1(code)
application Task2(code)
application Task3(code)
sector 2 data area
0x210000
physical sector 2
logical sector 2
application Task4(code)
sector 3 data area
0x300000
physical sector 16
logical sector 16
application TaskM
0x300000
physical sector 17
logical sector 17
application TaskM+1
physical sector 31
0x3fffff
ADDRESS OF RESET VECTOR 2=BOOT20
logical sector 0
boot20
copy_prog()
root Task()
downloader()
basic ISRs
logical sector 1
application Task1(code)
application Task2(code)
application Task3(code)
sector 2 data area

Fig. 8


ERROR PROCESSING START
READ OUT ADDRESS VALUE IN WHICH ERROR WAS GENERATED
S21
SPECIFY SECTOR IN WHICH ERROR WAS GENERATED FROM ADDRESS VALUE
S22
S23
ERROR IS GENERATED IN BOOT SECTOR?
NO
YES
S24
TRANSMIT REQUEST OF ALL DOWNLOAD
TRANSMIT REQUEST OF PARTIAL DOWNLOAD
S25
RETURN

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP98/00358

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[6] G06F9/06, 9/445, 12/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[6] G06F9/06, 9/445, 12/16

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1979-1998 Toroku Jitsuyo Shinan Koho 1994-1998
Kokai Jitsuyo Shinan Koho 1972-1994

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP, 8-69376, A (Hitachi, Ltd., Hitachi Keiyo Engineering K.K.), March 12, 1996 (12. 03. 96) (Family: none)	1-3, 7, 9, 10, 13, 14
Y		4-6, 8, 11, 12
Y	JP, 8-195952, A (Sony Corp.), July 30, 1996 (30. 07. 96) & EP, 723372, A2 & CN, 1138802, A	4
Y	JP, 7-261997, A (Fanuc Ltd.), October 13, 1995 (13. 10. 95) (Family: none)	5, 11
Y	WO, 95/08824, A1 (Robert Bosch GmbH.), March 30, 1995 (30. 03. 95) & DE, 4332499, A1 & CN, 1131997, A & US, 5712969, A & JP, 9-503083, A & EP, 721644, A1	5, 6

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
April 16, 1998 (16. 04. 98)	April 28, 1998 (28. 04. 98)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP98/00358

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP, 3-158184, A (Juki Corp.), July 8, 1991 (08. 07. 91) (Family: none)	8
Y	JP, 5-197559, A (NEC Corp.), August 6, 1993 (06. 08. 93) (Family: none)	12
A	JP, 8-255084, A (Hewlett-Packard Co.), October 1, 1996 (01. 10. 96) & EP, 723226, A1 & US, 5568641, A	1-14

Form PCT/ISA/210 (continuation of second sheet) (July 1992)